FIG. 1

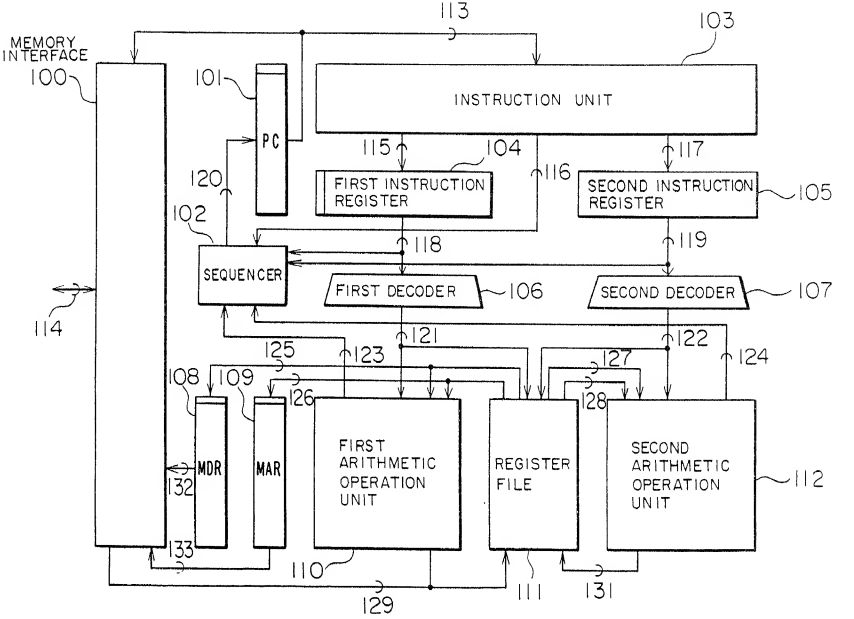


FIG. 2
PRIOR ART

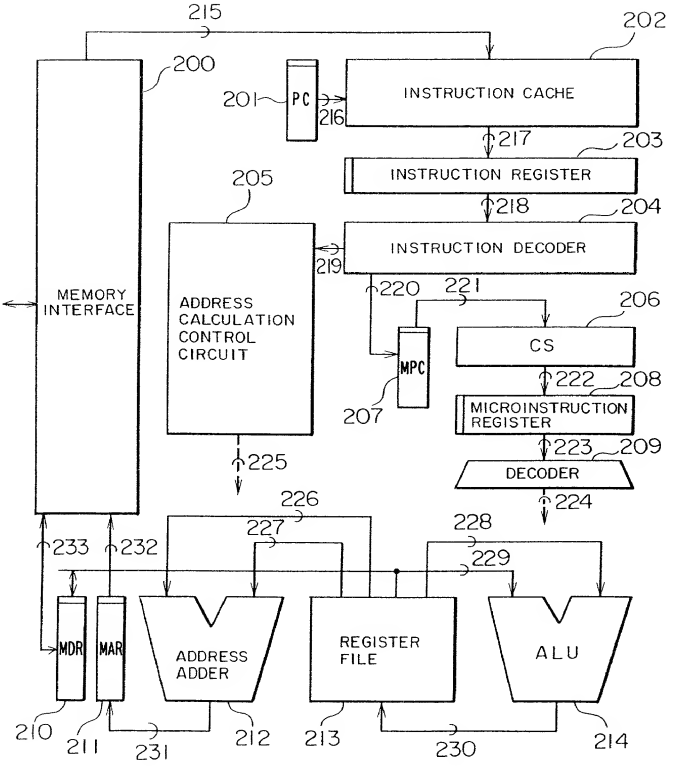


FIG. 3
PRIOR ART

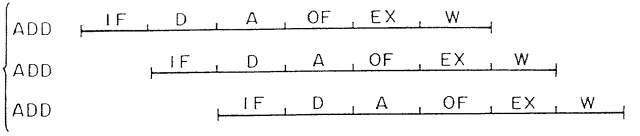


FIG. 4
PRIOR ART

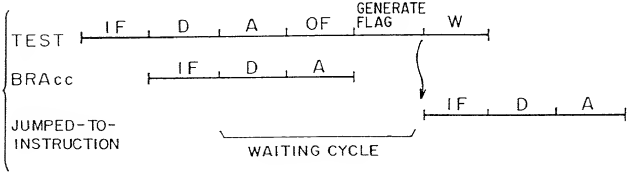


FIG. 5
PRIOR ART

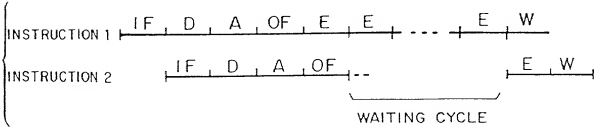


FIG. 6
PRIOR ART

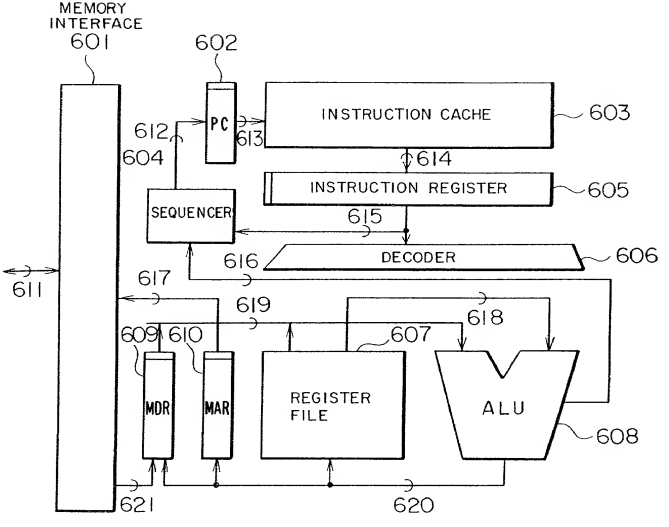


FIG. 7
PRIOR ART

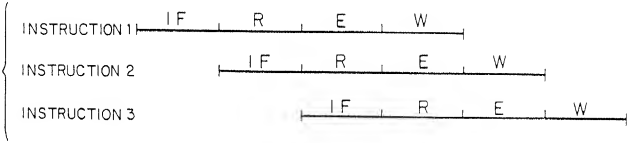


FIG. 8
PRIOR ART

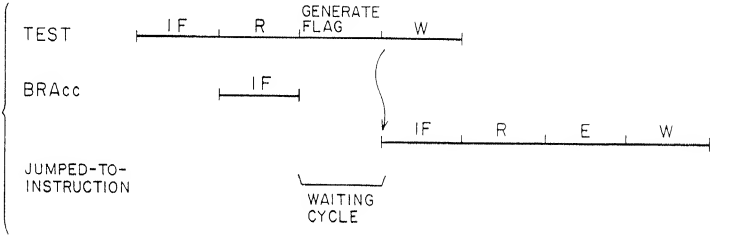


FIG. 9

TYPES	MNEMONIC	OPERATION
BASIC INSTRUCTION	ADD R(S1), R(S2), R(D)	R(S1)+R(S2)→R(D)
	SUB "	R(S1)−R(S2)→R(D)
	AND "	STORE LOGICAL PRODUCT OF EACH BITS OF R(S1), R(S2) IN R(D)
	OR "	STORE LOGICAL SUM OF EACH BITS OF R(S1), R(S2) IN R(D)
	EOR "	STORE EXCLUSIVE OR OF EACH BITS OF R(S1), R(S2) IN R(D)
	NOT R(S1), R(D)	STORE LOGICAL NOT OF EACH BIT OF R(S1) IN R(D)
	SFT R(S1), R(S2), R(D)	SHIFT R(S1) BY BIT NUMBER INDICATED BY R(S2) AND STORE IN R(D)
	NOP	DO NOTHING
BRANCH INSTRUCTION	BRA d	PC+d→PC
	BRAcc d	
	CALL d	PC→R(O), PC+d→PC
	RTN d	R(O)→PC
LOAD STORE INSTRUCTION	STOR R(S1), R(S2)	WRITE R(S1) IN MEMORY POINTED BY R(S2)
	LOAD R(S1), R(D)	WRITE DATA OF MEMORY POINTED BY R(S1) IN R(D)

FIG. 10

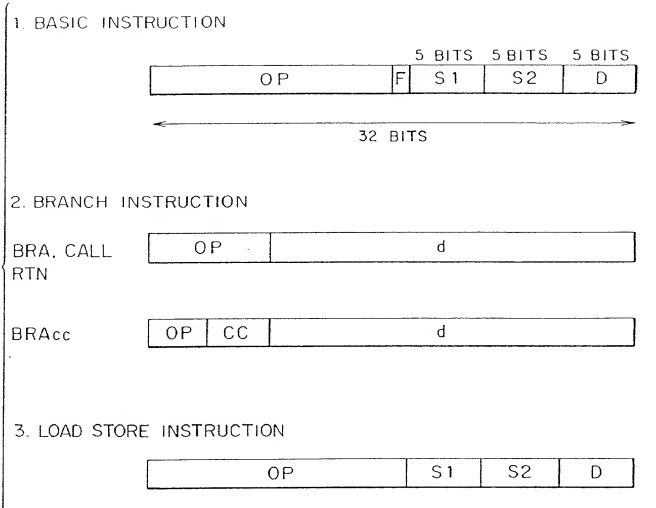

1. BASIC INSTRUCTION
5 BITS
5 BITS
5 BITS
OP
F
S1
S2
D
32 BITS
2. BRANCH INSTRUCTION
BRA, CALL
RTN
OP
d
BRAcc
OP
CC
d
3. LOAD STORE INSTRUCTION
OP
S1
S2
D

FIG. 11

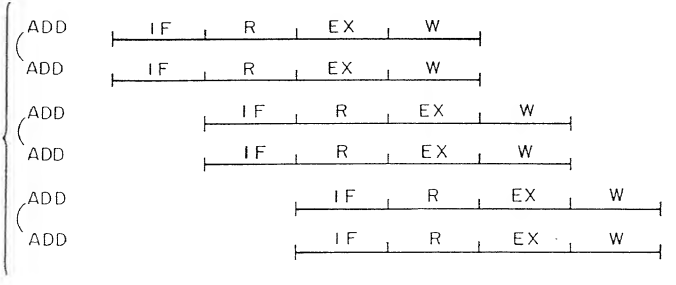

ADD
ADD
ADD
ADD
ADD
ADD
IF
R
EX
W

FIG. 12

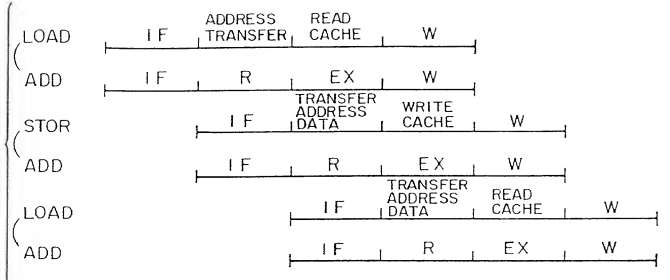

LOAD
ADD
STOR
ADD
LOAD
ADD
IF
ADDRESS TRANSFER
READ CACHE
R
EX
W
TRANSFER ADDRESS DATA
WRITE CACHE

FIG. 13

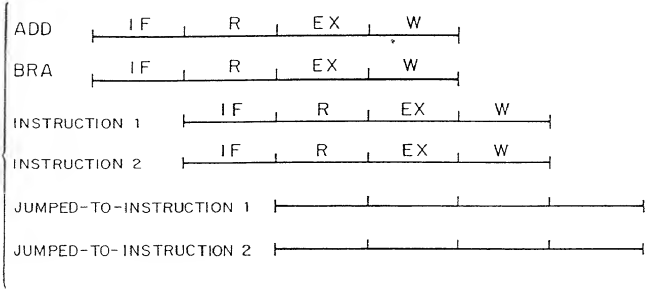

ADD
BRA
INSTRUCTION 1
INSTRUCTION 2
JUMPED-TO-INSTRUCTION 1
JUMPED-TO-INSTRUCTION 2
IF
R
EX
W

FIG. 14

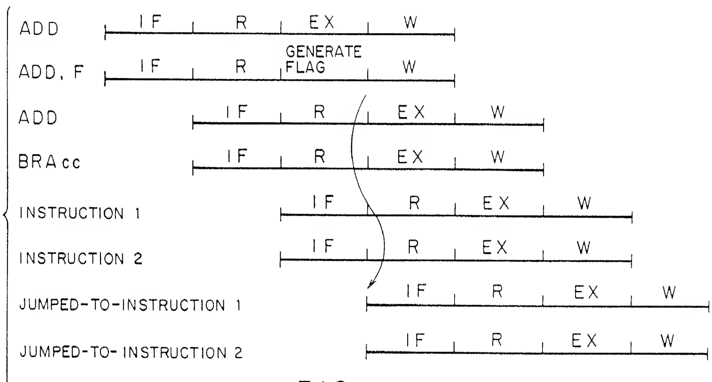


FIG. 15 PRIOR ART

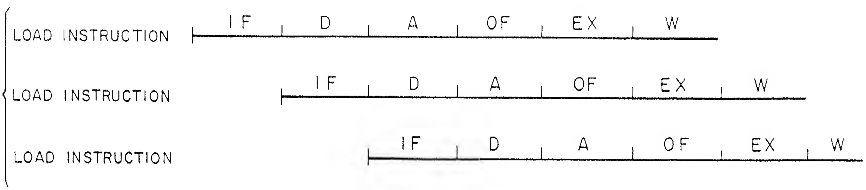


FIG. 16

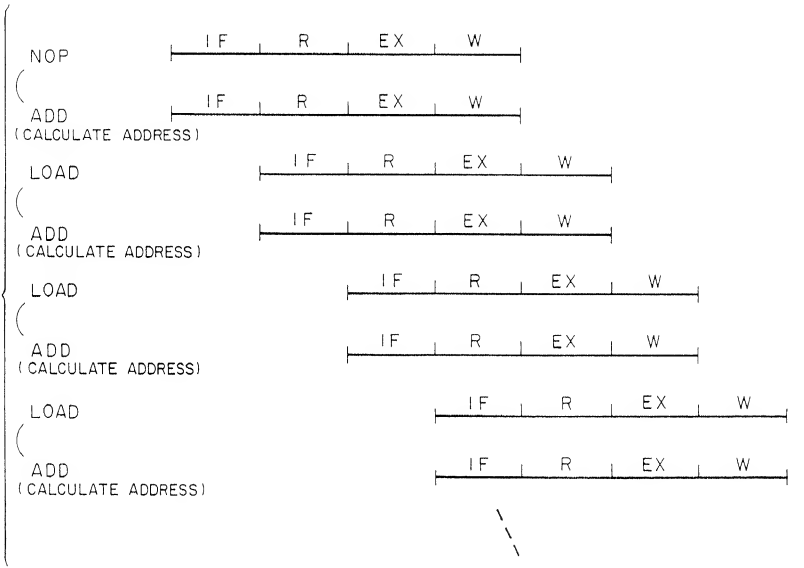


FIG. 17.
PRIOR ART

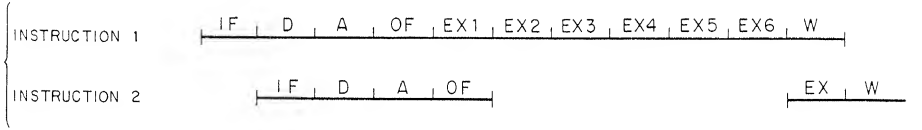


FIG. 18

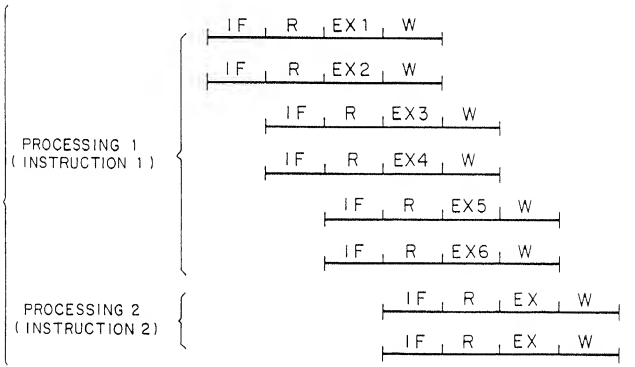


FIG. 19

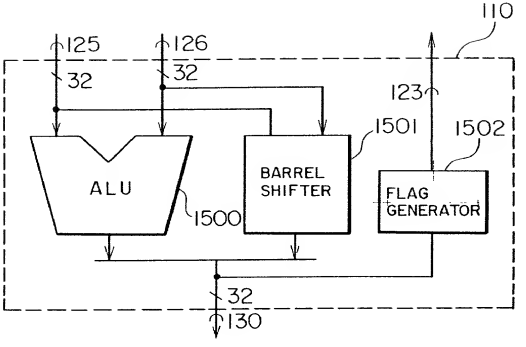

110
125
126
32
32
123
1501
1502
ALU
1500
BARREL SHIFTER
FLAG GENERATOR
32
130

FIG. 20

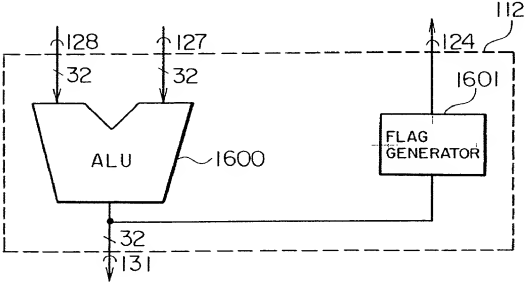
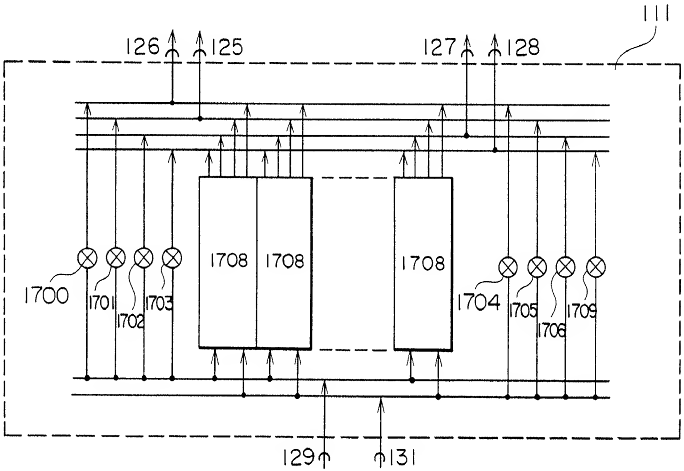

112
128
127
124
32
32
1601
ALU
1600
FLAG GENERATOR
32
131

FIG. 21



111
126
125
127
128
1708
1708
1708
1700
1701
1702
1703
1704
1705
1706
1709
129
131

FIG. 22

ADDRESS	FIRST INSTRUCTION			SECOND INSTRUCTION		
0	SFT	R(1),	R(2), R(3)	ADD	R(4), R(5),	R(6)
2	SFT	R(7),	R(8), R(9)	SFT	R(10), R(11),	R(12)
4	ADD	R(14),	R(15), R(16)	ADD	R(17), R(18),	R(19)
			↓			
PC						
0	SFT	R(1),	R(2), R(3)	ADD	R(4), R(5),	R(6)
2	SFT	R(7),	R(8), R(9)	NOP		
3	SFT	R(10),	R(11), R(12)	NOP		
4	ADD	R(14),	R(15), R(16)	ADD	R(17), R(18),	R(19)

FIG. 23

ADDRESS						
0	SFT	R(1),	R(2), R(3)	ADD	R(4), R(5),	R(6)
2	SFT	R(7),	R(8), R(9)	NOP		
4	SFT	R(10),	R(11), R(12)	NOP		
6	ADD	R(14),	R(15), R(16)	ADD	R(17), R(18),	R(19)

FIG. 24

ADDRESS	FIRST INSTRUCTION	SECOND INSTRUCTION
0	ADD R(1), R(2), R(3)	ADD R(4), R(5), R(6)
2	LOAD R(3), R(10)	LOAD R(6), R(11)
4	ADD R(5), R(2), R(3)	ADD R(4), R(1), R(6)

⇩

PC	FIRST INSTRUCTION	SECOND INSTRUCTION
0	ADD R(1), R(2), R(3)	ADD R(4), R(5), R(6)
2	LOAD R(3), R(10)	NOP
3	LOAD R(6), R(11)	NOP
4	ADD R(5), R(2), R(3)	ADD R(4), R(1), R(6)

FIG. 25

ADDRESS	FIRST INSTRUCTION	SECOND INSTRUCTION
0	ADD R(1), R(2), R(3)	ADD R(4), R(5), R(6)
2	ADD R(1), R(5), R(8)	ADD R(8), R(9), R(10)
4	ADD R(12), R(13), R(14)	ADD R(15), R(16), R(17)

⇩

PC	FIRST INSTRUCTION	SECOND INSTRUCTION
0	ADD R(1), R(2), R(3)	ADD R(4), R(5), R(6)
2	ADD R(1), R(5), R(8)	NOP
3	ADD R(8), R(9), R(10)	NOP
4	ADD R(12), R(13), R(14)	ADD R(15), R(16), R(17)

FIG. 26

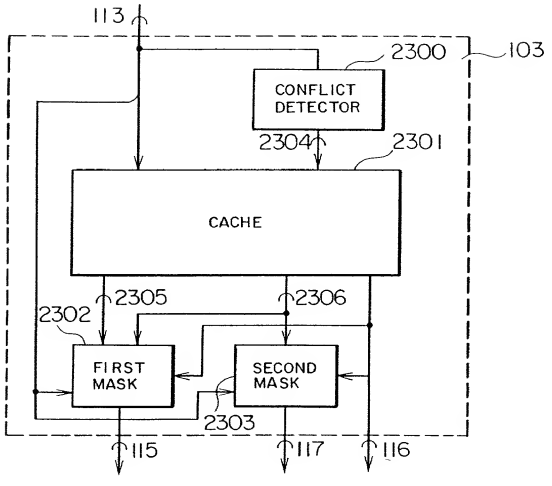


FIG. 27

CONFLICT BIT	LSB OF PC	FIRST INSTRUCTION SIGNAL 115	SECOND INSTRUCTION SIGNAL 117
0	0	FIRST INSTRUCTION	SECOND INSTRUCTION
0	1	NOP	SECOND INSTRUCTION
1	0	FIRST INSTRUCTION	NOP
1	1	SECOND INSTRUCTION	NOP

FIG. 28

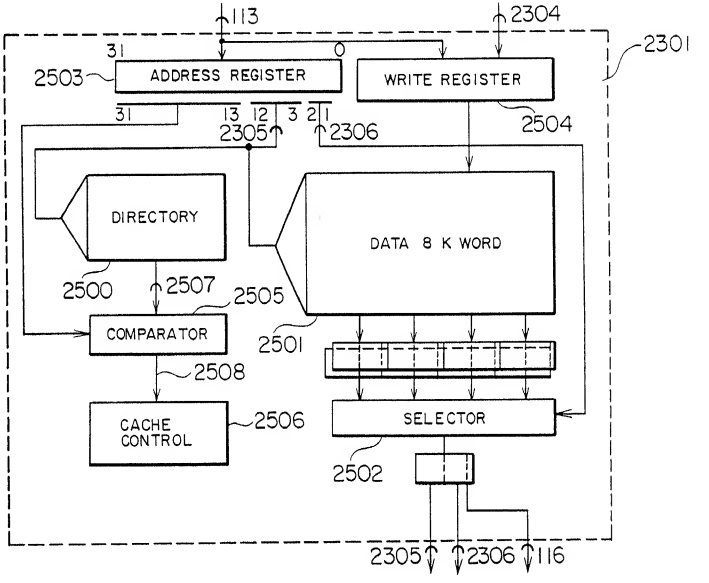


FIG. 29

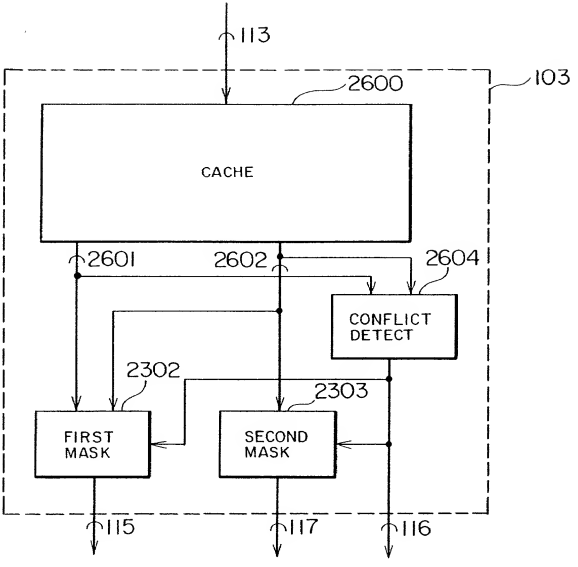

113
103
2600
CACHE
2601
2602
2604
CONFLICT DETECT
2302
2303
FIRST MASK
SECOND MASK
115
117
116

FIG. 30

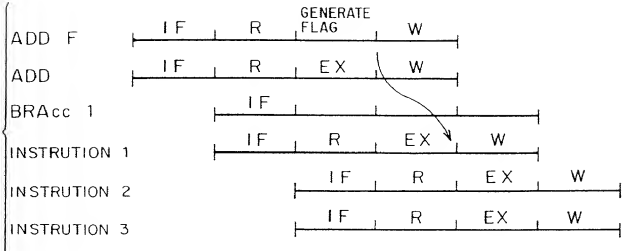


FIG. 31A
PRIOR ART

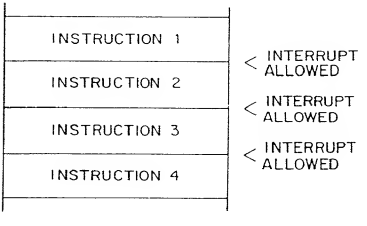


FIG. 31B

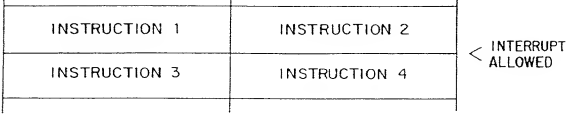


FIG. 32

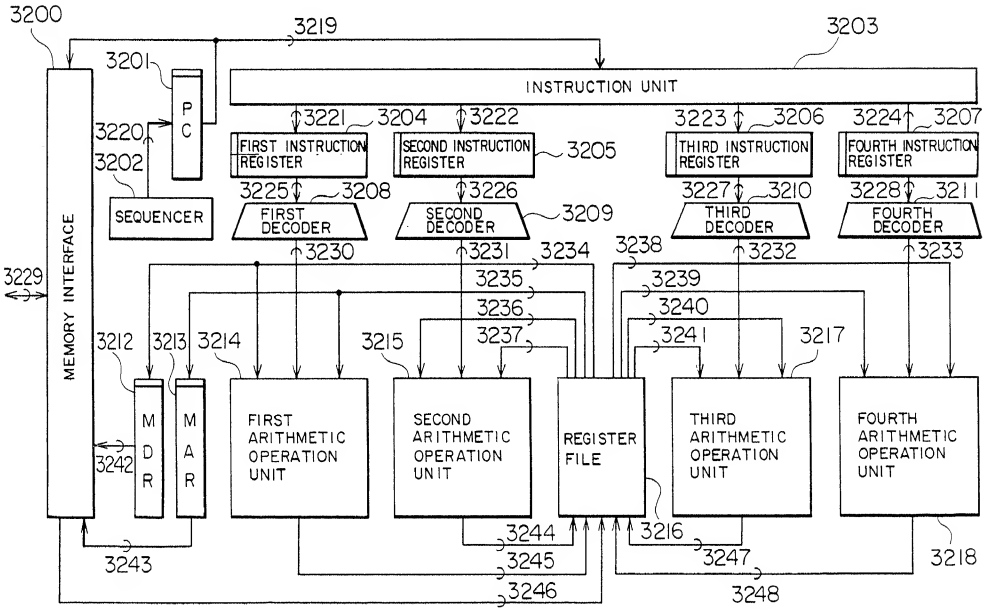


FIG. 33(a)

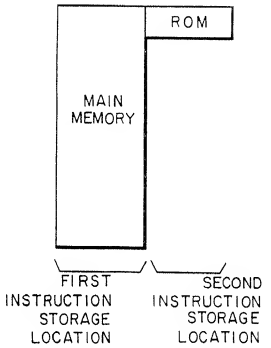


FIG. 33(b)

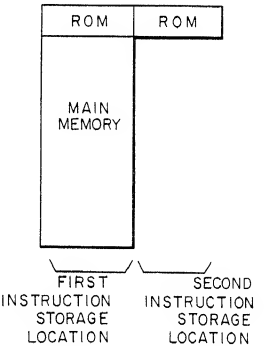


FIG. 33(c)

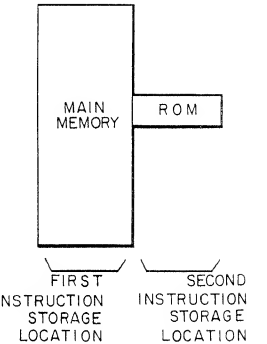


FIG. 34

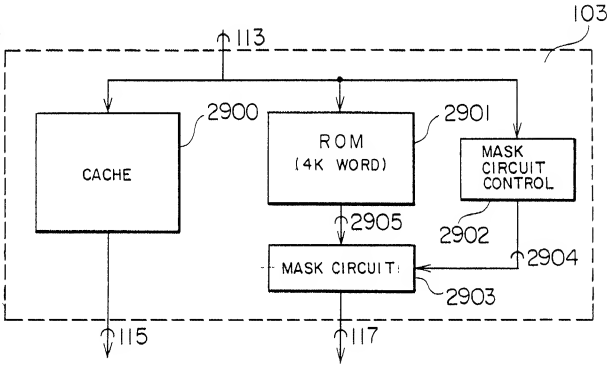


FIG. 35

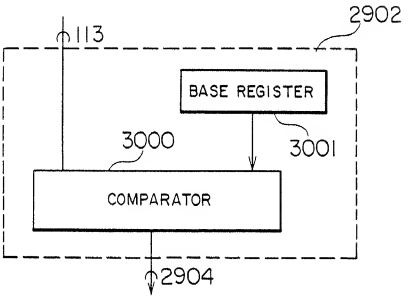
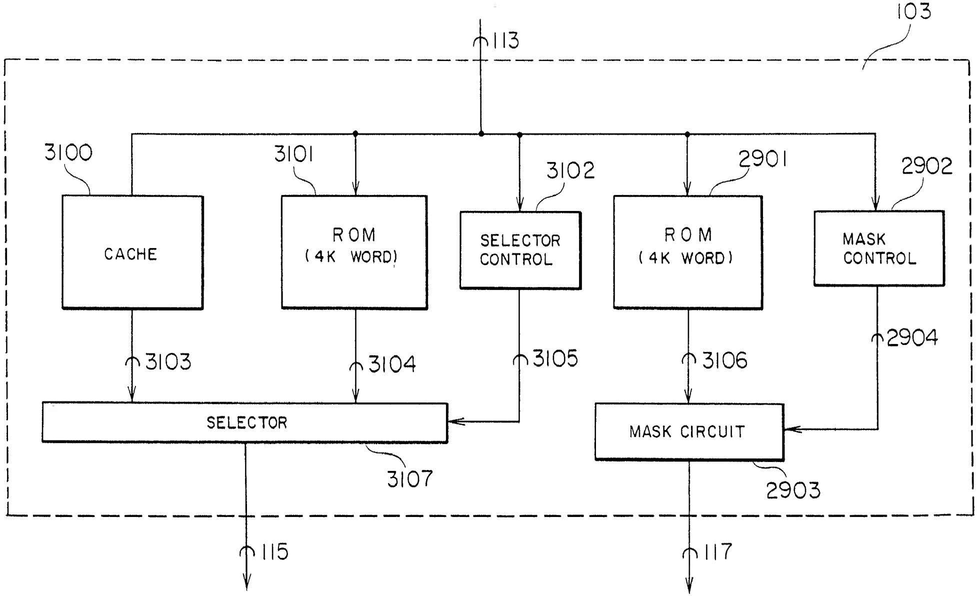


FIG. 36


103
113
3100
CACHE
3101
ROM
(4K WORD)
3102
SELECTOR
CONTROL
2901
ROM
(4K WORD)
2902
MASK
CONTROL
3103
3104
3105
3106
2904
SELECTOR
MASK CIRCUIT
3107
2903
115
117